

Business Processes Automation

D. Vanderbist - 3/10/2016

Content

1. RPA vs BPMS: Characteristics and Trade-Offs
2. BPMS State of Technology: Orchestration vs. BPMS



RPA vs. BPMS

Characteristics and Trade-Offs

RPA vs. BPMS (1/3)

RPA:

Products:

- BluePrism, Automation Anywhere, Power Automate

Typical Use-Case:

- To integrate LOB systems **through the UI** when there is no means to get information from the LOB system through a system-to-system interface

Characteristics:

- Typically **less building blocks** and out-of-the-box integration components to build the full business process A-Z.
- Process **complexity is limited** to flow chart like flows. Not a lot of support for hierarchical or nested processes.
- **Processes are executed atomically from begin to end.** Limited or no support for long-running processes that can be interrupted mid execution. The duration of an activity is at the level of magnitude of seconds.

BPMS:

Products:

- K2, AgilePoint, Windows Workflow Manager

Typical Use-Case:

- The process is composed of activities executed in LOB system (CRM, ERP) and they can be targeted through **system-to-system integration**
- The process requires human intervention and can be targeted to a **human-to-system integration**: writing a custom UI to handle the human input and link the UI through normal integration strategies with the process (web-service, messaging, DB) => **Important: this human interaction is not UI-integration => SEE RPA**

Characteristics:

- Typically used when there are a **lot of activities in the process and some complex logic to drive the process**
- **Processes can be interrupted mid execution (long running)** waiting for an activity to complete. This interruption can be at the level of magnitude of hours, days, months, years



RPA vs. BPMS (2/3)

RPA:

Characteristics:

- **Elaborate components to integrate LOB system at screen level:** so screen scraping (visual pixel level) or screen spying (API widget ID level)
- Typically used when there are limited and simple activities in the process to automate, the process is organized to overcome the UI integration => **Important this UI integration is not human-process interactions => SEE BPMS**
- **Hard to make abstraction of the workflow and the systems** it integrated with as UI is used for integration. This tight coupling through the UI requires a new flow per system that is integrated..

BPMS:

Characteristics:

- Typically have **no components to integrate LOB system at screen level:** so screen scraping (visual pixel level) or screen spying (API widget ID level)
- **Supports abstraction of the workflow from the systems it integrates with** as technical interfaces are used for the integration (API/Web services). Loose coupling is used and can reuse flows as long as the system respect the same technical interface. This abstraction (partner management) is typically done by using middleware like BizTalk.

RPA vs. BPMS (3/3)

RPA:

Gaps in supporting all types of business processes and all levels of complexity.

Good components for UI integration

BPMS:

Gaps in UI integration

Good support for all complex long running business processes.



So RPA-BPMS hybrid?

Pro's and Con's:

- **Disadvantage of hybrid:** two licenses and two products
- **Advantages of hybrid:** optimizing the advantages of RPA and BPMS

Trade-off factors to make decision:

- **Is UI-integration** required and can no alternatives be found (through web-services, files, messages, DB)? Go for RPA
- Is the **business process complex**? Go for BPMS

Trade-Offs and Other Technologies

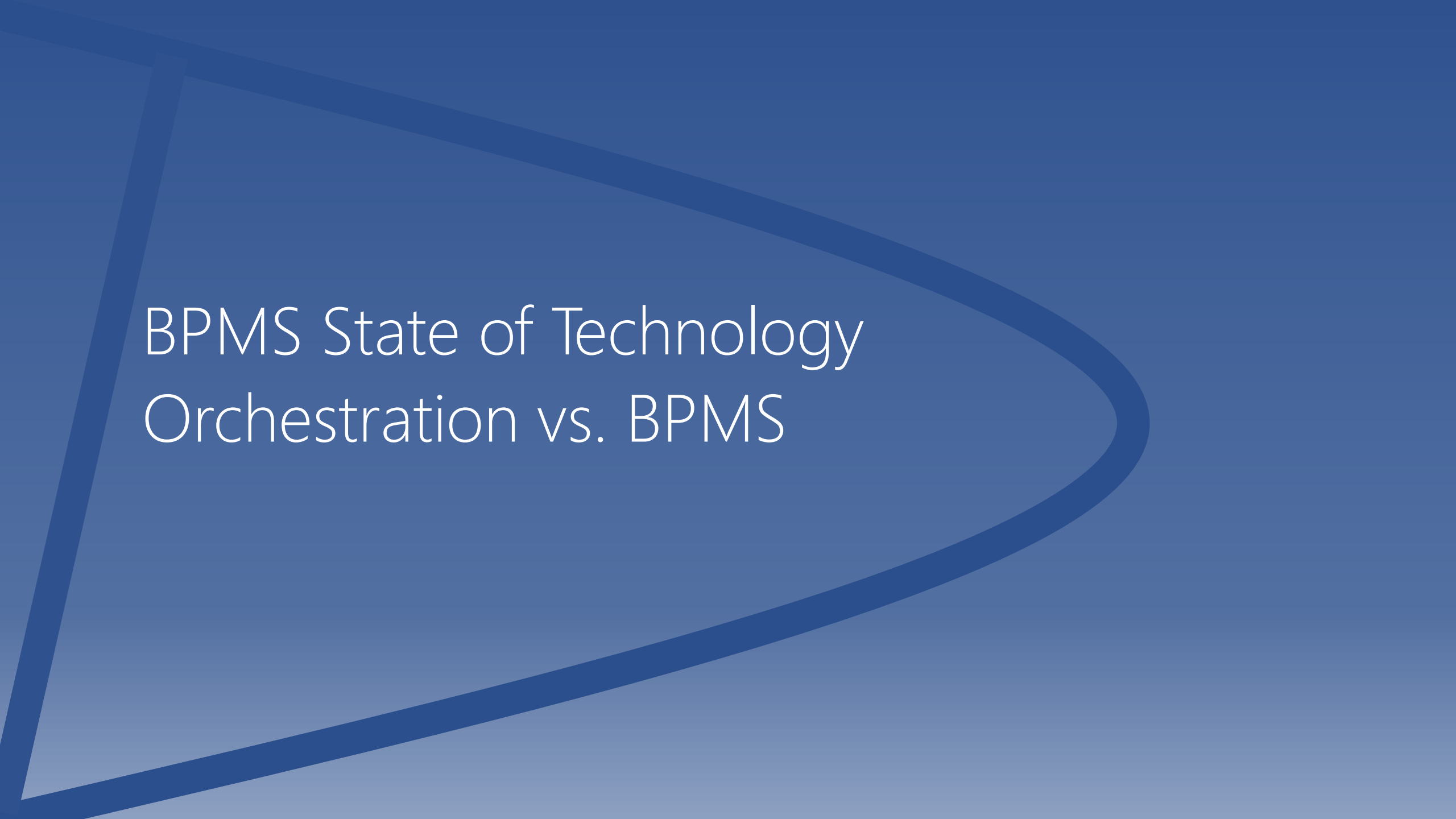
Related technologies and questions:

OCR vs. RPA: do not abuse RPA system for OCR scanning!

Orchestrations vs. BPMS: atomic processes vs interruptible processes.

Communication Bus (Messaging) vs. Orchestrator; atomic-requests vs. unit-of-work requests





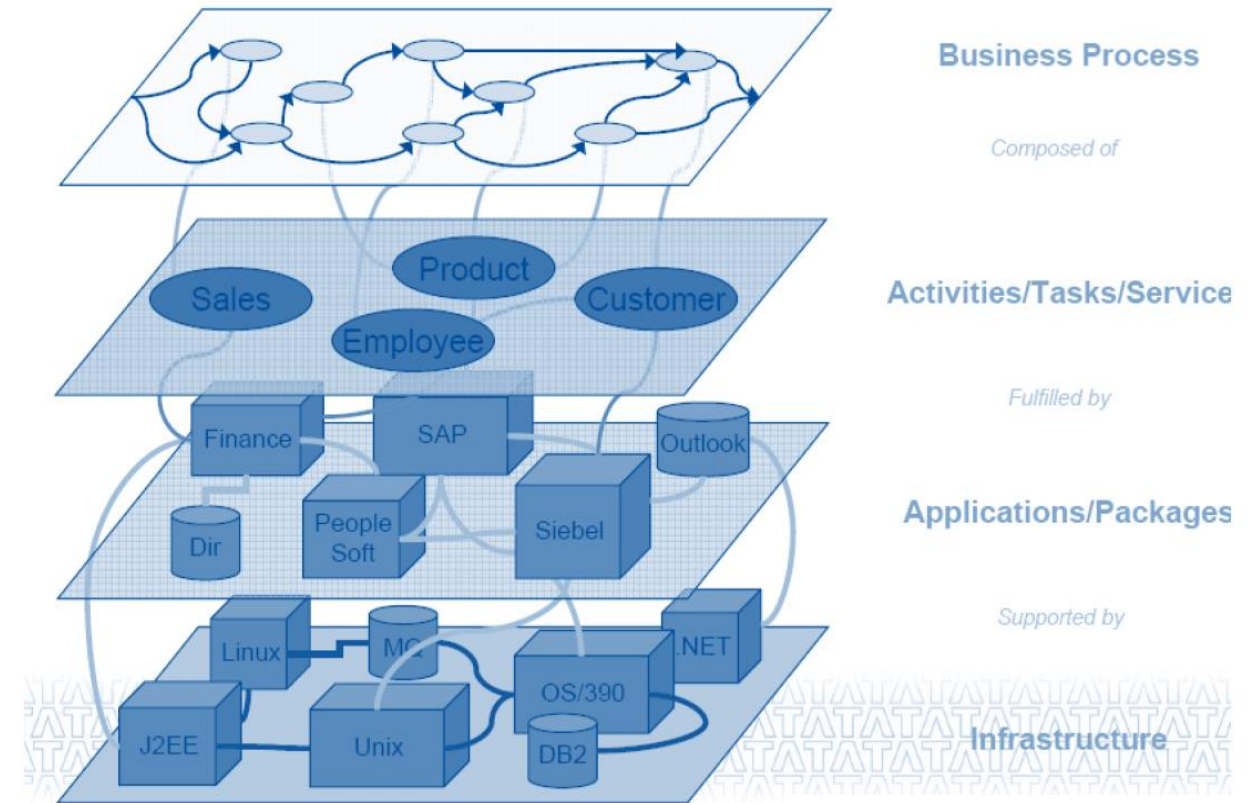
BPMS State of Technology Orchestration vs. BPMS

Process Technology

Introduction

Process automation is solved through **enterprise integration architectures** to realize **business processes** with following functional characteristics:

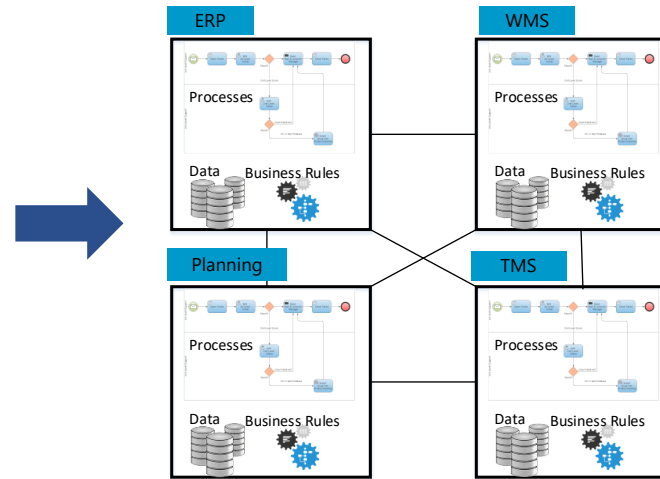
- **Multiple systems** need to be combined to realize the overall process
- **Multiple data sources** are required e.g. reference data and transactional data
- The overall **process** has a certain **degree of change over time**
- **Multiple versions** of a **business process** must be supported at given moment in time.
- The **process activities** can have **fixed or variable behavior**
- The **process activities** can be executed by **systems** or humans
- **Business intelligence** can initiate **changes** in the overall process or process activities
- An insight, **end-to-end-view**, in the overall status of the process and the activities is required



State of Technology

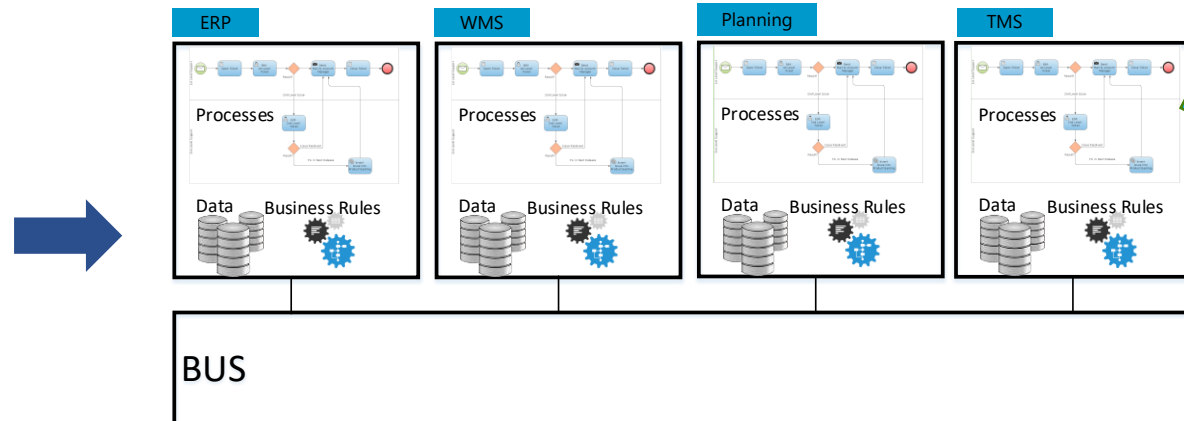
Process Integration 1/7

1. Separated Back-Ends – Ad-Hoc:



Data Oriented Integration

2. Communication Integrated – Communication Bus:



Data Oriented Integration

State of Technology

Process Integration 2/7

Data Oriented Integration

Data-Oriented Application Architectures:

Functional Data Domains:

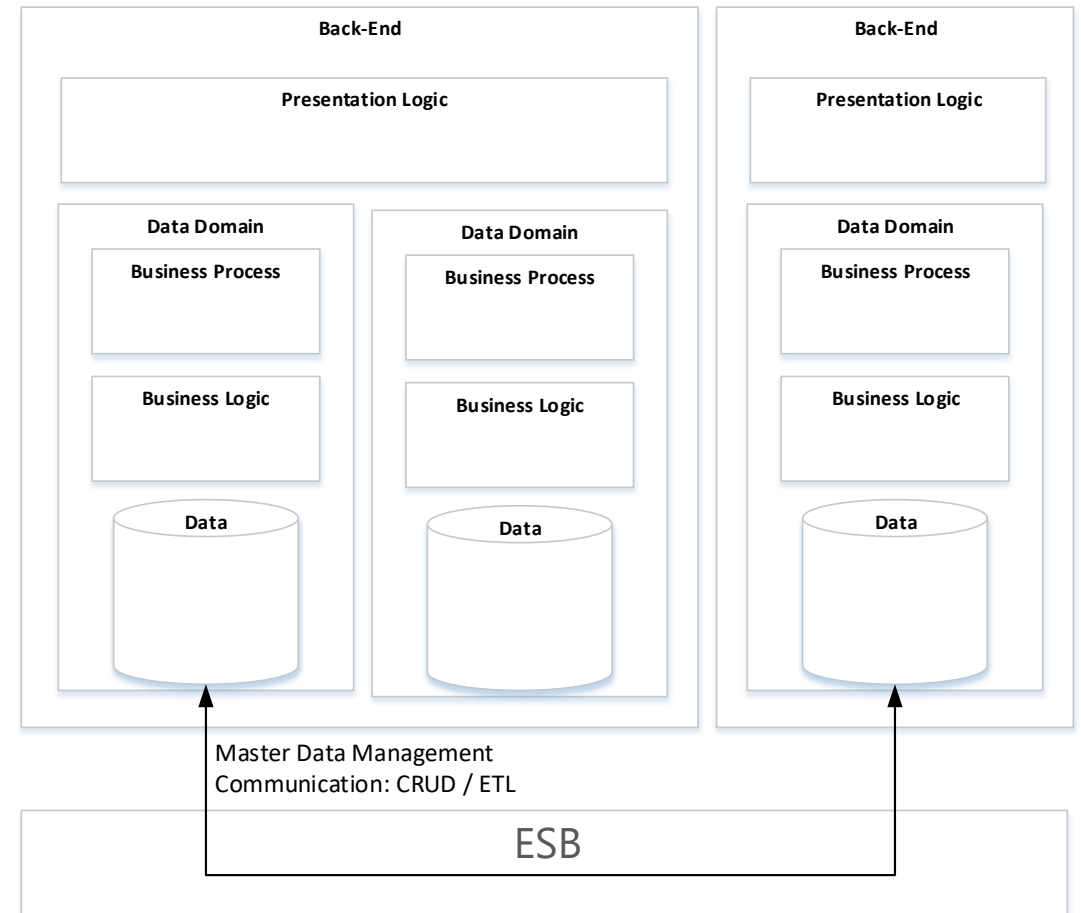
- One application is owner of a specific object and encapsulates the business logic of the object
- One domain per department
- Isolated data silo's

Data Exchange:

- Real-time: CRUD
- Batch: ETL

Business Processes:

- Limited to a department
- Cross-departmental process view require data to be centralized or replicated



State of Technology

Process Integration 3/7

Communication Integrated – Communication Bus:

- Integration style: using a common EAI technology (typically SOA & ESB)
- Business logic and processes: distributed over back-end application
- Process management: processes managed within one back-end

Data Oriented Integration

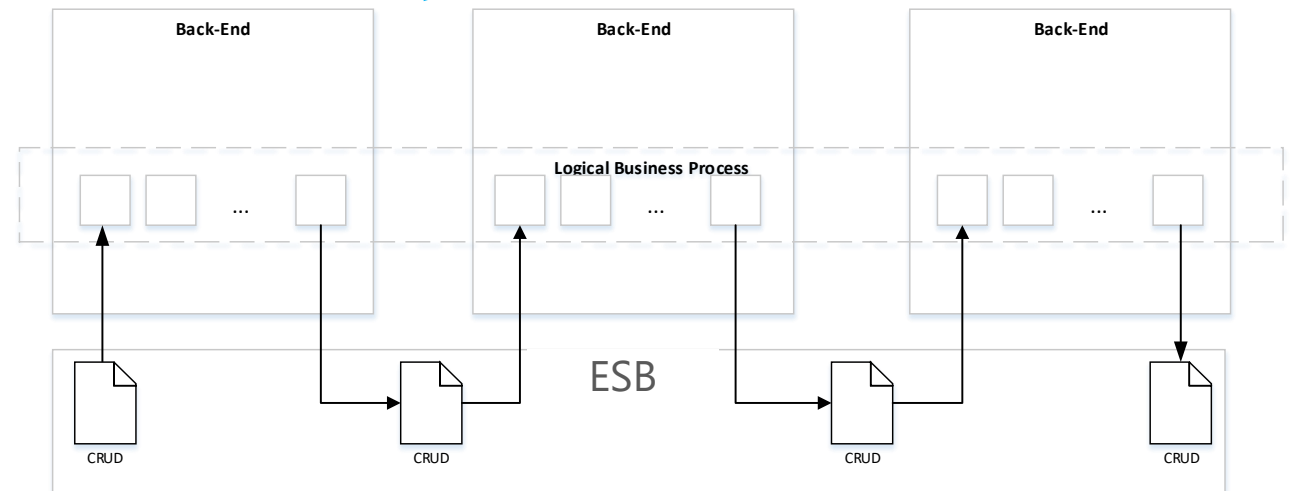
Data-Oriented Application Architectures:

Cross Department Processes

- Communication across ESB
- Process steps in multiple back-ends
- Back-End applications move data to next application in line
- Sender has knowledge about next application which results in tight-coupling

Business process characteristics:

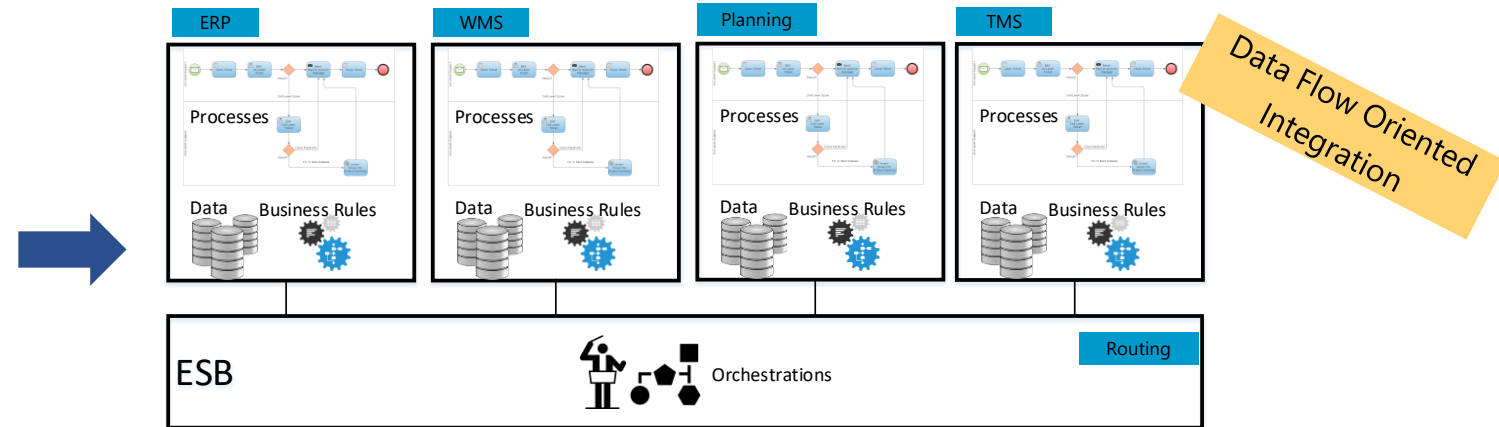
- Multiple systems and multiple data sources are supported.
- The systems and data sources are stable over time.
- The process is linear, stable over time and process activities can change within the scope of one back-end system.
- Multiple versions of a process must be handled by the individual back-end systems in each back-system participation in the process.
- No human-process interaction or human interaction can be organized in one of the back-end systems.
- The process logic is pre-defined and does change to real-time events.
- An end-to-end view can be organized within one of the participating back-end systems.



State of Technology

Process Integration 4/7

3. Process Integrated – Orchestrations:



State of Technology

Process Integration 5/7

Process Integrated – Orchestrations:

- Integration style: using a common EAI technology (typically SOA & ESB)
- Business logic and processes: distributed over back-end application
- Process management: cross back-end system processes piloted from one back-end system, event-based state managed using orchestrations

Data Flow Oriented Integration

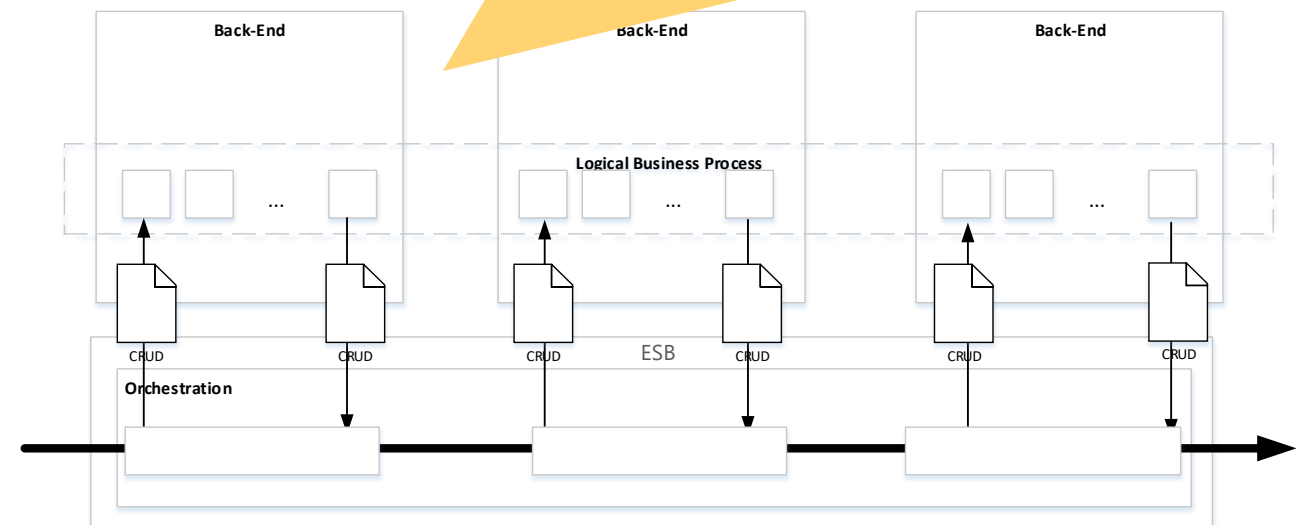
Data-Flow Oriented Application Architectures:

Orchestrations in ESB:

- Process driven by the back-end applications
- Loosely coupled: orchestration routes and delivers data across systems, sender has no knowledge about next application in line
- Not a real isolated end-to-end process: long running process build from short running processes in back-ends
- Stateless: state is encapsulated in the data communicated between system (transient data), orchestration itself has no state (condition was relaxed over time)
- Advantage: back-end application are the best equipped to deal with there own data

Business process characteristics:

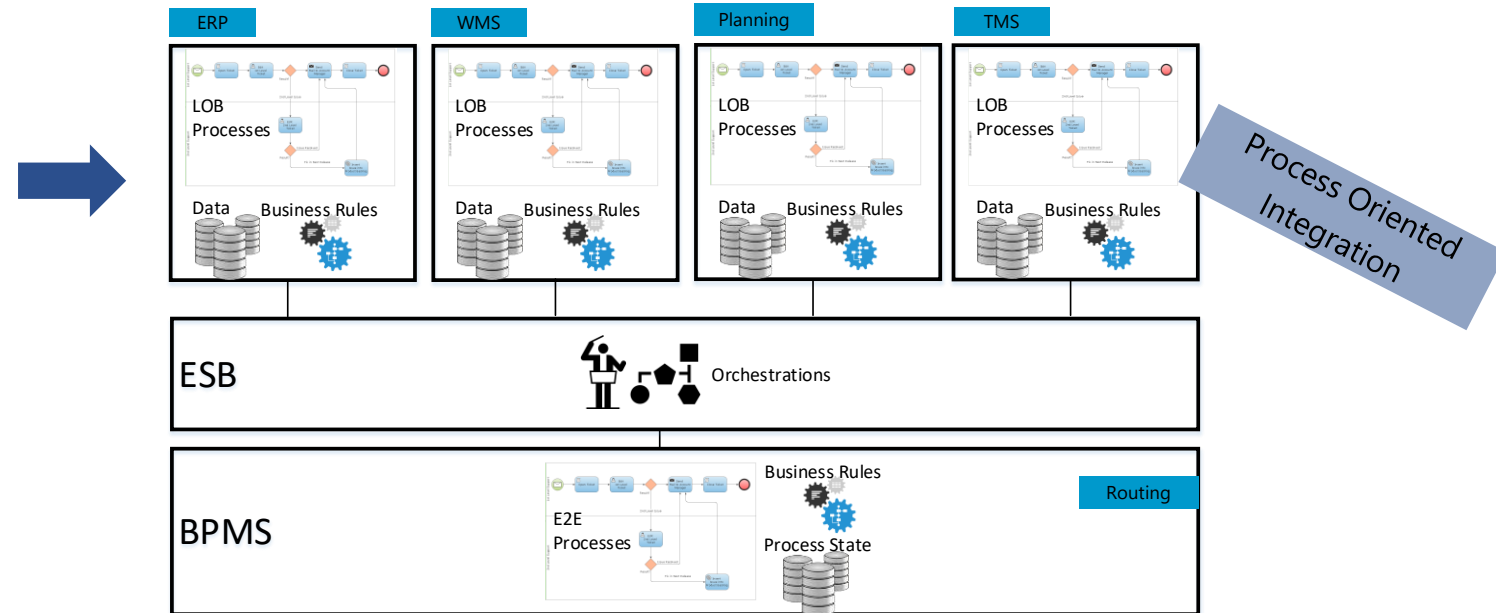
- Multiple systems and multiple data sources are supported.
- Systems and data sources can change over time.
- The process can easily be extended to new systems and can support non-linear flows.
- Multiple versions of a process can be handled by the orchestration engine as long as back-end system provide the correct data. Started process can't change and will run to completion.
- Process activities can change within the scope of one back-end system.
- No human-process interaction or human interaction can be organized in one of the back-end systems.
- Process are considered to be atomic at the Orchestration level: short life-span and non-interruptible.
- Process logic is pre-defined and does change to real-time events but can use common rules at orchestration level.
- An end-to-end view can be organized at the orchestration level.



State of Technology

Process Integration 6/7

4. Process Integrated - BPMS:



State of Technology

Process Integration 7/7

Process Integrated - BPMS:

- Integration style: using a common EAI technology
- Business logic and processes: centralized processes using logic of back-end application
- Process management: end-to-end processes piloted from BPMS system, process state managed in BPMS

Process Oriented Integration

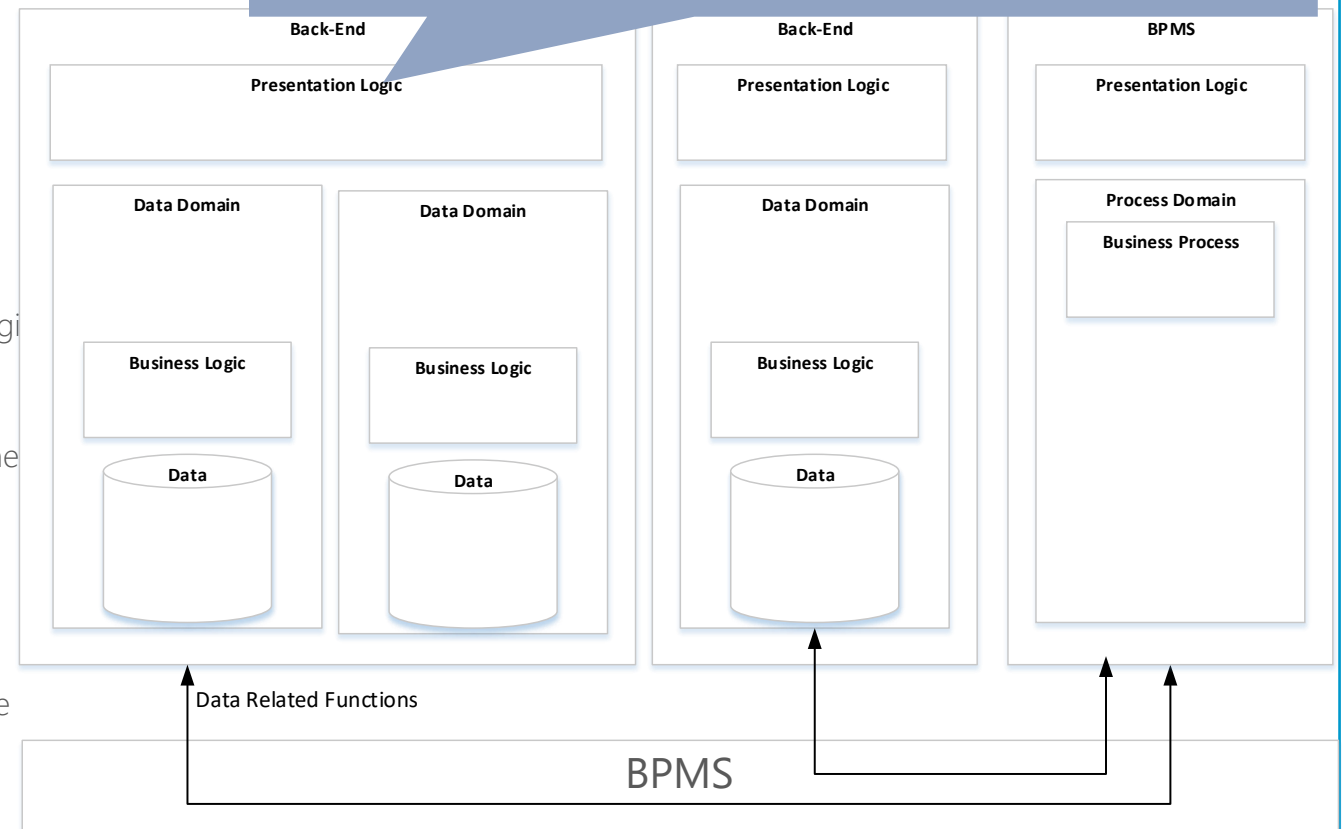
Process Oriented Application Architectures:

Processes run in the BPMS:

- Process driven by the BPMS
- Loosely coupled: only the BPMS is aware of steps in the process logic
- Isolated: end-to-end processes in the BPMS, logic is called on the back-end systems to participate in processes
- Function oriented: instead of CRUD (data access) operations, business function are called
- State-full:
 - Data that triggered the process: payload
 - Original context the process was created in
 - Process state
- Advantage: process logic is centralized/isolated so it can be flexible changed, business logic remains in the back-end applications

Business process characteristics:

- Multiple systems and multiple data sources are supported.
- Systems and data sources can change over time.
- The process can easily be extended to new systems and can support non-linear flows.
- Multiple versions of a process can be handled by the BPMS engine as long as back-end system provide the correct data. Started process can change version while running if required or can run to completion,
- Process activities are organized in the BPMS and can change whenever required.
- Human-process interaction or human interaction is organized at BPMS level having it own UI when necessary.
- Process are considered long-running, can be interrupted or can be idle. Process logic can react to real-time events and can use common business rules.
- An end-to-end view can be organized at the BPMS level.



Some Architecture Trade-Offs

Data & Data-Flow vs. Process

- Speed vs. Isolated Changes
- Centrally managed vs. Distributed

Questions to guide decisions between Orchestrations and BPMS:

- Are multiple versions of a process required at the same time?
- Is a process atomic or interruptible?
- Is human intervention required and organized in a central UI?
- Will the process activities change outside the scope of the back-end systems?
- Will the process require real-time behavioral change based on alerts?

Orchestration Engine:

- Single Version
- **Atomic**
- No Human Interaction
- Change scope at back-end systems
- No real-time process change

BPMS:

- Multi-Version
- Interruptible
- Human Interaction
- Change scope at process
- Real-time process change

Implementing Processes

Introduction

Enterprise Integration Architecture:

- Business logic driven
 - Communication Bus:
 - **Data oriented** architecture
 - To realize a custom developed router
- Business process driven
 - Orchestration Engine
 - **Data flow** oriented architecture
 - Applied in **automated process split** across applications
 - Business Process Management System
 - **Process oriented** architectures
 - Applied in **automated processes externalize** from applications



a.k.a. Option 1



a.k.a. Option 2

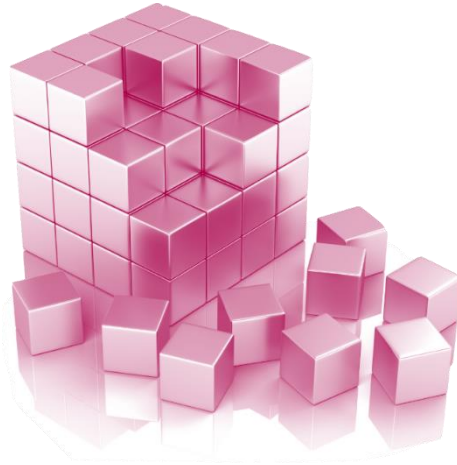


a.k.a. Option 3

Option 1 - Custom Development based



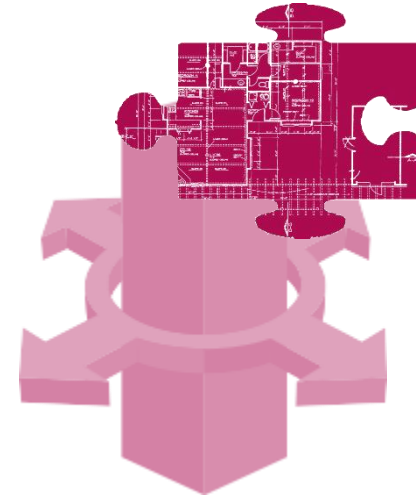
Fully customized
for optimal
performance



No out-of-the-
box building
blocks



Requires full unit,
component and
integration tests
(V-Model)

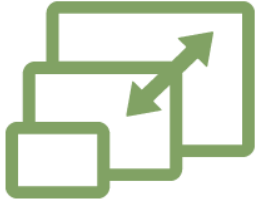


Performance
dependent on
correct design
(Scaling)



Quality
dependent on
team strength

Option 2 - Orchestration Engine based

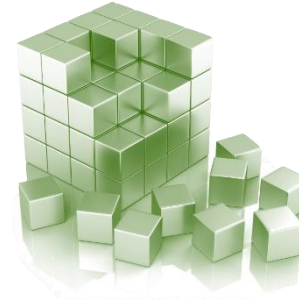


Allows for fast response times and process execution

(given ERP, TPM, WMS,, BI follow the request speed/load)



Easily extensible with new CC's, DC's, PC's, ...

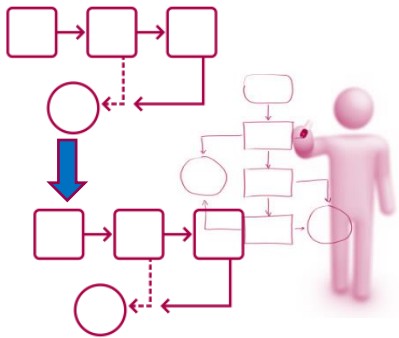


Out-of-the-box building blocks like rule engine, decision tables

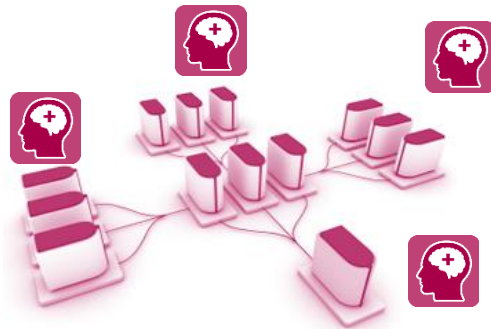


Can manage relationships between PO, SO, Preq ... other documents

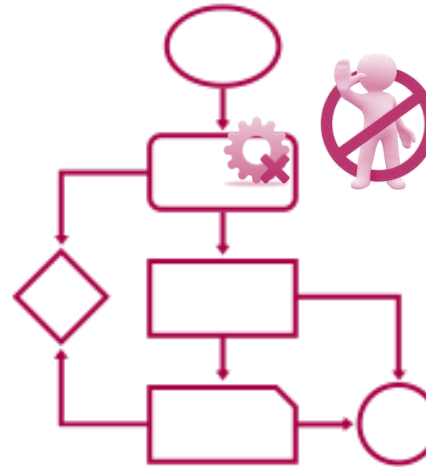
- requires less custom development work so less risk and costs
- is more cost effective as not a full BPMS is rolled out for only one process



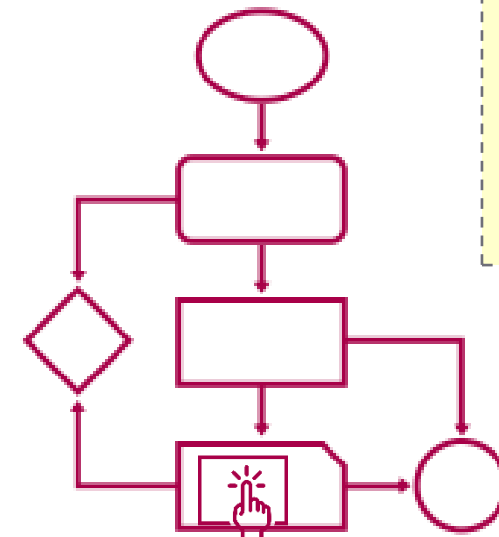
No support for interaction between processes or multiple versions of a process



The process has no "memory" itself, must be organized by the participating components (ERP, TPM, WMS, ...)

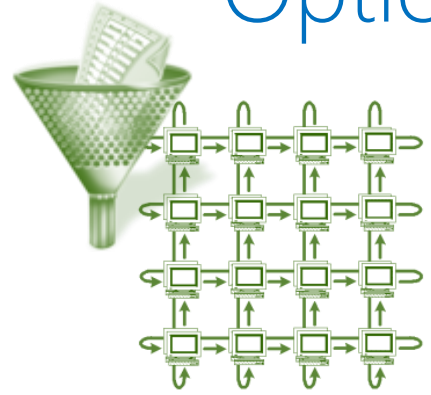


The process can not be interrupted midst execution

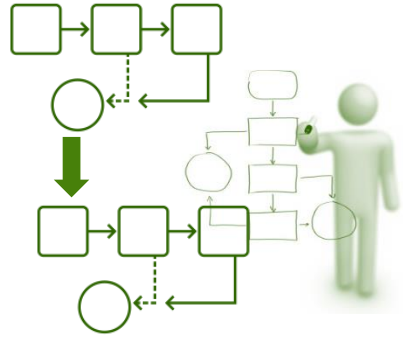


The process does not support human interaction

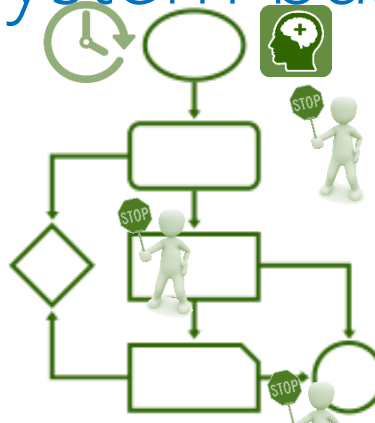
Option 3 - BPMS system based



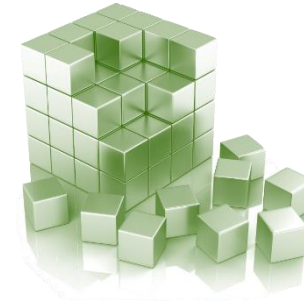
Allows for large volume of processes executed at the same time (at the expense of longer response times)



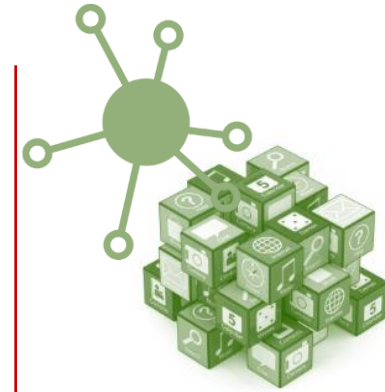
Supports interaction between processes, or multiple versions of a process and changes to processes



Process can be interrupted and has a memory (process state)



Out-of-the-box building blocks like rule engine, decision tables

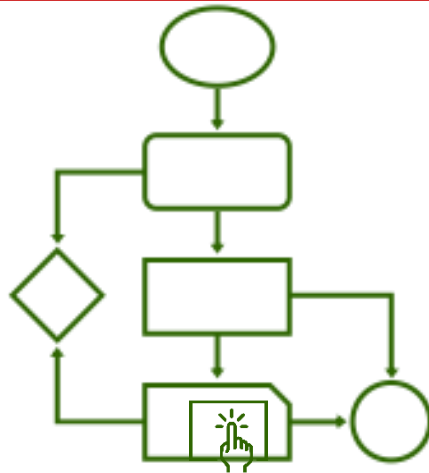


Out-of-the-box connectors for well known systems (SAP, AX, Salesforce ...)

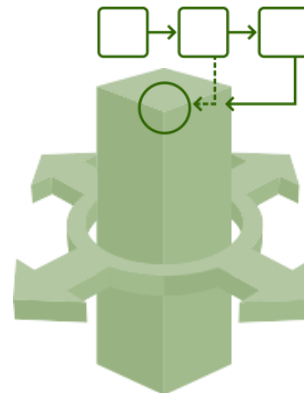
- offers the most flexibility for the future
- is a bit over-architecture for only running one process



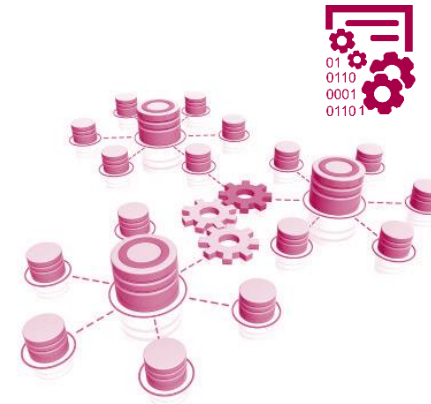
Process simulation is often supported



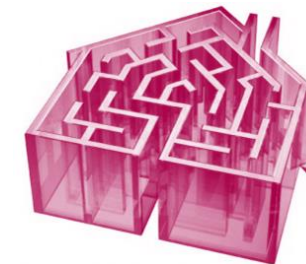
The process supports human interaction through BPMS UI components



Equipped to support many different processes and running process instances



Less efficient in data oriented architecture



Rather expensive for the implementation of one process